Workshop on the ACTS Collection

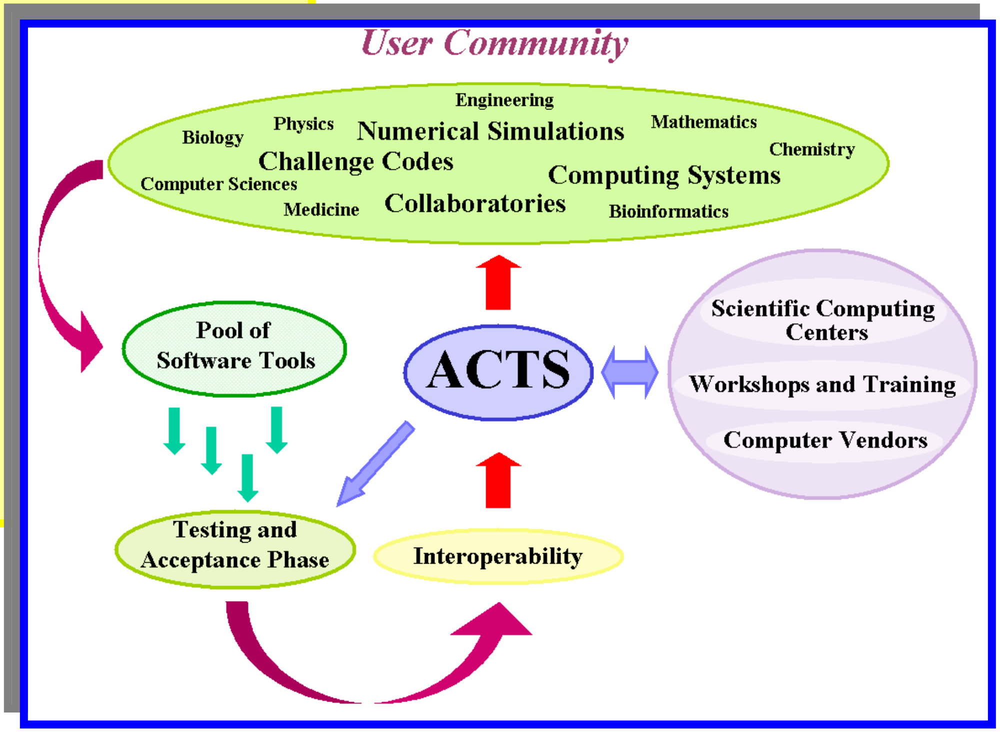
Enabling Technologies For High End Computer Simulations

August 24-27, 2004

Osni Marques
Lawrence Berkeley National Laboratory
oamarques@lbl.gov

Outline

- Keeping the pace with the software and hardware
  - Hardware evolution
  - Performance tuning
  - Software selection
  - Other challenges
- The DOE ACTS Collection Project
  - Goals
  - Current features
  - A sample of scientific applications.
  - Why do we need all these tools?
  - Lessons learned



What is the ACTS Collection?

http://acts.nersc.gov

- **A**dvanced **C**ompu**T**ational **S**oftware Collection
- Tools for developing parallel applications
- ACTS started as an "umbrella" project

Goals

- *Extended support for experimental software*
- *Make ACTS tools available on DOE computers*
- *Provide technical support (acts-support@nersc.gov)*
- *Maintain ACTS information center (http://acts.nersc.gov)*
- *Coordinate efforts with other supercomputing centers*
- *Enable large scale scientific applications*
- *Educate and train*

- ***High***
  - Intermediate level
  - Tool expertise
  - Conduct tutorials
- ***Intermediate***
  - Basic level
  - Higher level of support to users of the tool
- ***Basic***
  - Help with installation
  - Basic knowledge of the tools
  - Compilation of user's reports

ACTS Tools Functionalities

Category	Tool	Functionalities
Numerical	**Aztec**	Algorithms for the iterative solution of large sparse linear systems.
	Hypre	Algorithms for the iterative solution of large sparse linear systems, intuitive grid-centric interfaces, and dynamic configuration of parameters.
	PETSc	Tools for the solution of PDEs that require solving large-scale, sparse linear and nonlinear systems of equations.
	OPT++	Object-oriented nonlinear optimization package.
	SUNDIALS	Solvers for the solution of systems of ordinary differential equations, nonlinear algebraic equations, and differential-algebraic equations.
	ScaLAPACK	Library of high performance dense linear algebra routines for distributed-memory message-passing.
	SuperLU	General-purpose library for the direct solution of large, sparse, nonsymmetric systems of linear equations.
	TAO	Large-scale optimization software, including nonlinear least squares, unconstrained minimization, bound constrained optimization, and general nonlinear optimization.
Code Development	**Global Arrays**	Library for writing parallel programs that use large arrays distributed across processing nodes and that offers a shared-memory view of distributed arrays.
	Overture	Object-Oriented tools for solving computational fluid dynamics and combustion problems in complex geometries.
Code Execution	**CUMULVS**	Framework that enables programmers to incorporate fault-tolerance, interactive visualization and computational steering into existing parallel programs
	Globus	Services for the creation of computational Grids and tools with which applications can be developed to access the Grid.
	PAWS	Framework for coupling parallel applications within a component-like model.
	SILOON	Tools and run-time support for building easy-to-use external interfaces to existing numerical codes.
	TAU	Set of tools for analyzing the performance of C, C++, Fortran and Java programs.
Library Development	**ATLAS and PHiPAC**	Tools for the automatic generation of optimized numerical software for modern computer architectures and compilers.
	PETE	Extensible implementation of the expression template technique (C++ technique for passing expressions as function arguments).

$Ax = b$

$Az = \lambda z$

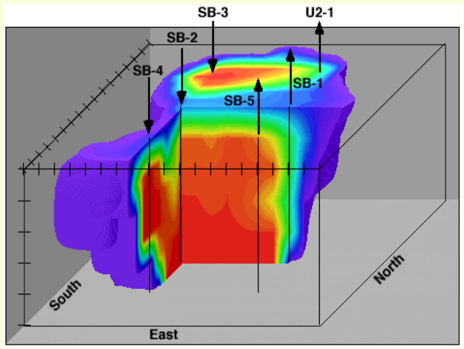
$A = U\Sigma V^T$

PDEs

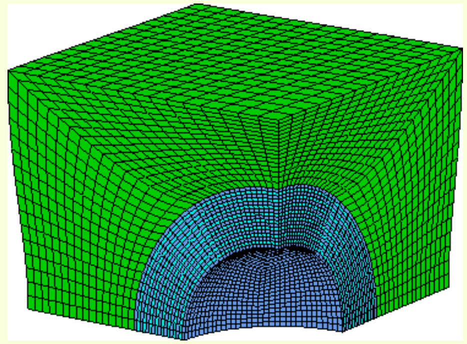
ODEs

$\vdots$

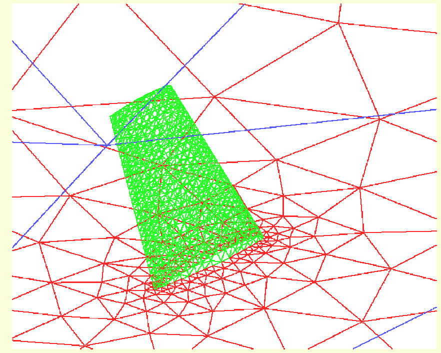
Use of ACTS Tools



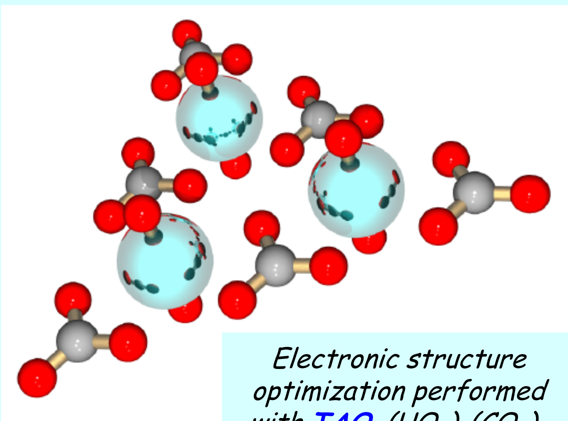
Multiphase flow using ***PETSc****, 4 million cell blocks, 32 million DOF, over 10.6 Gflops on an IBM SP (128 nodes), entire simulation runs in less than 30 minutes (Pope, Gropp, Morgan, Seperhrnoori, Smith and Wheeler).*



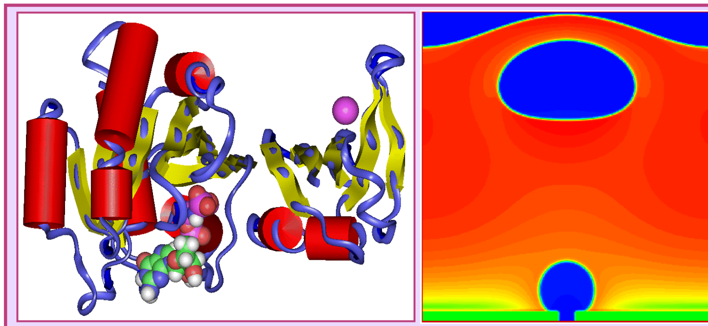
Model of a "hard" sphere included in a "soft" material, 26 million d.o.f. Unstructured meshes in solid mechanics using Prometheus and ***PETSc*** *(Adams and Demmel).*



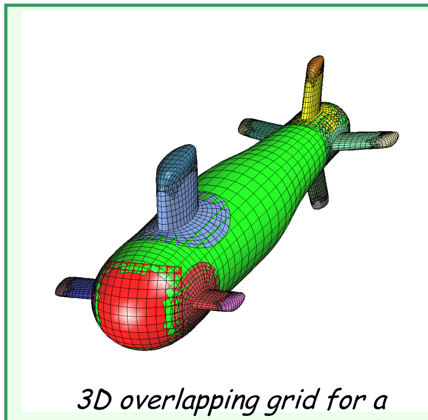
3D incompressible Euler,tetrahedral grid, up to 11 million unknowns, based on a legacy NASA code, FUN3d (W. K. Anderson), fully implicit steady-state, parallelized with ***PETSc*** *(courtesy of Kaushik and Keyes).*



Electronic structure optimization performed with ***TAO****, $(UO_2)_3(CO_3)_6$ (courtesy of deJong).*

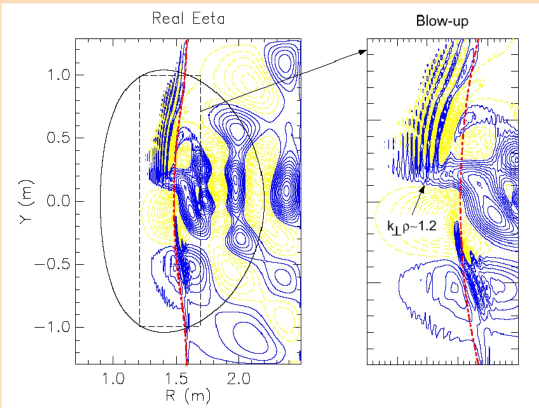


Molecular dynamics and thermal flow simulation using codes based on ***Global Arrays****. GA have been employed in large simulation codes such as NWChem, GAMESS-UK, Columbus, Molpro, Molcas, MWPhys/Grid, etc.*

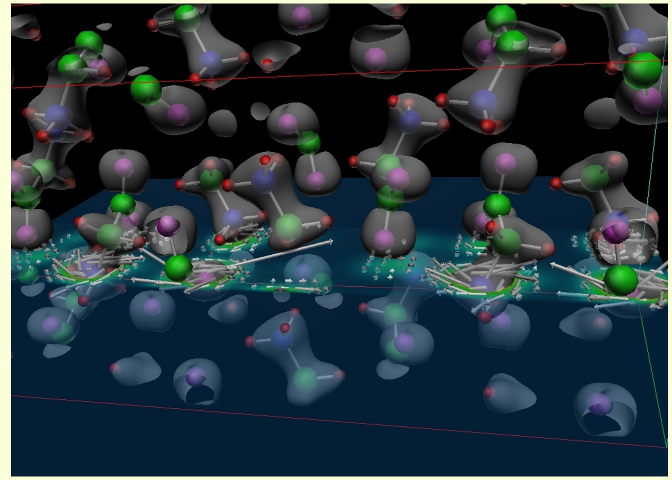


3D overlapping grid for a submarine produced with ***Overture****'s module ogen.*

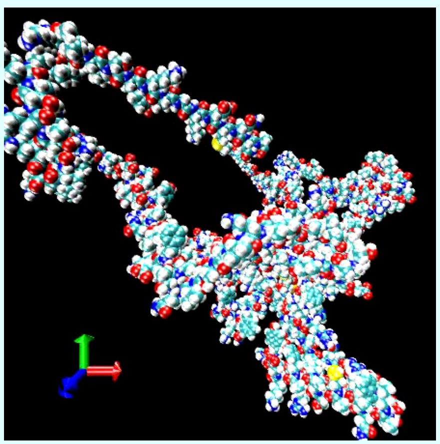
Use of ACTS Tools



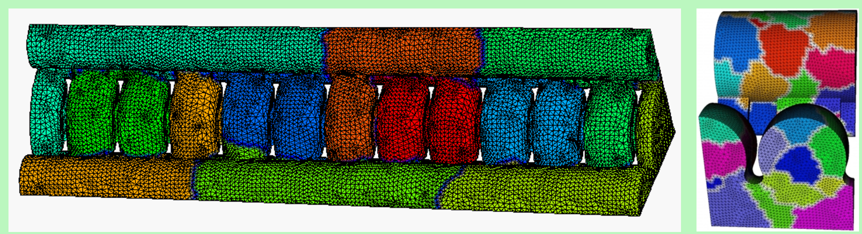
*Two **ScaLAPACK** routines, PZGETRF and PZGETRS, are used for solution of linear systems in the spectral algorithms based AORSA code (Batchelor et al.), which is intended for the study of electromagnetic wave-plasma interactions. The code reaches 68% of peak performance on 1936 processors of an IBM SP.*



*Induced current (white arrows) and charge density (colored plane and gray surface) in crystallized glycine due to an external field (Louie, Yoon, Pfrommer and Canning), eigenvalue problems solved with **ScaLAPACK**.*



***OPT++** is used in protein energy minimization problems (shown here is protein T162 from CASP5, courtesy of Meza , Oliva et al.)*



*Omega3P is a parallel distributed-memory code intended for the modeling and analysis of accelerator cavities, which requires the solution of generalized eigenvalue problems. A parallel exact shift-invert eigensolver based on PARPACK and **SuperLU** has allowed for the solution of a problem of order 7.5 million with 304 million nonzeros. Finding 10 eigenvalues requires about 2.5 hours on 24 processors of an IBM SP.*

High Performance Computers (Sustainable Performance)

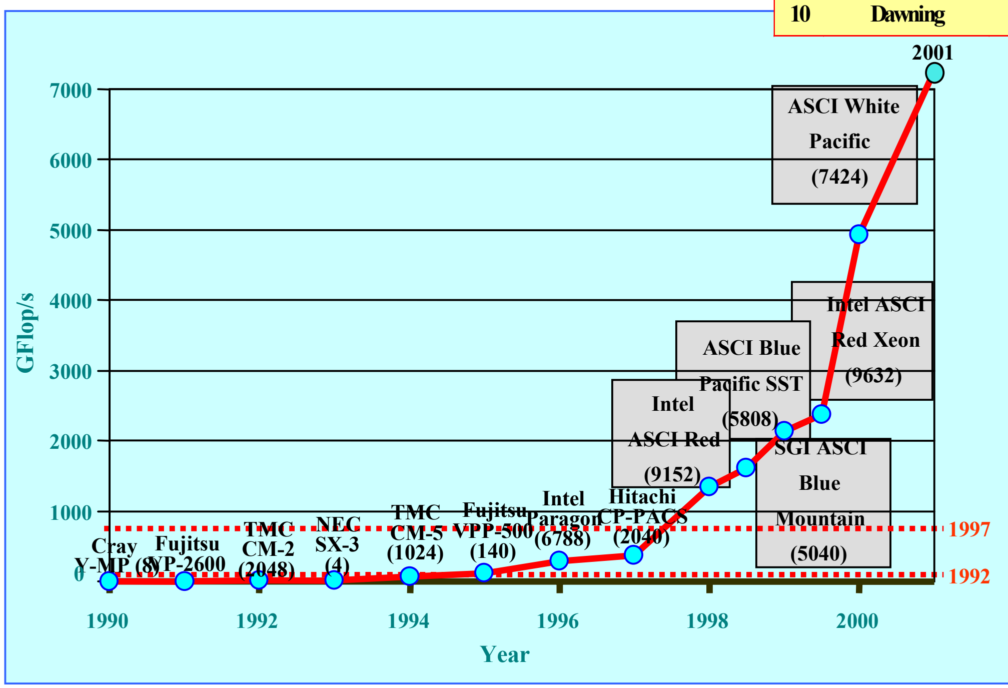
- ~ 20 years ago → $1x10^6$ Floating Point Ops/sec (Mflop/s)
  - Scalar based
- ~ 10 years ago → $1x10^9$ Floating Point Ops/sec (Gflop/s)
  - Vector & Shared memory computing, bandwidth aware
  - Block partitioned, latency tolerant
- ~ Today → $1x10^{12}$ Floating Point Ops/sec (Tflop/s)
  - Highly parallel, distributed processing, message passing, network based
  - data decomposition, communication/computation
- Coming soon → $1x10^{15}$ Floating Point Ops/sec (Pflop/s)
  - Many more levels of memory hierarchy, combination of grids&HPC
  - More adaptive, latency and bandwidth aware, fault tolerant, extended precision, attention to SMP nodes

What does that performance evolution mean?

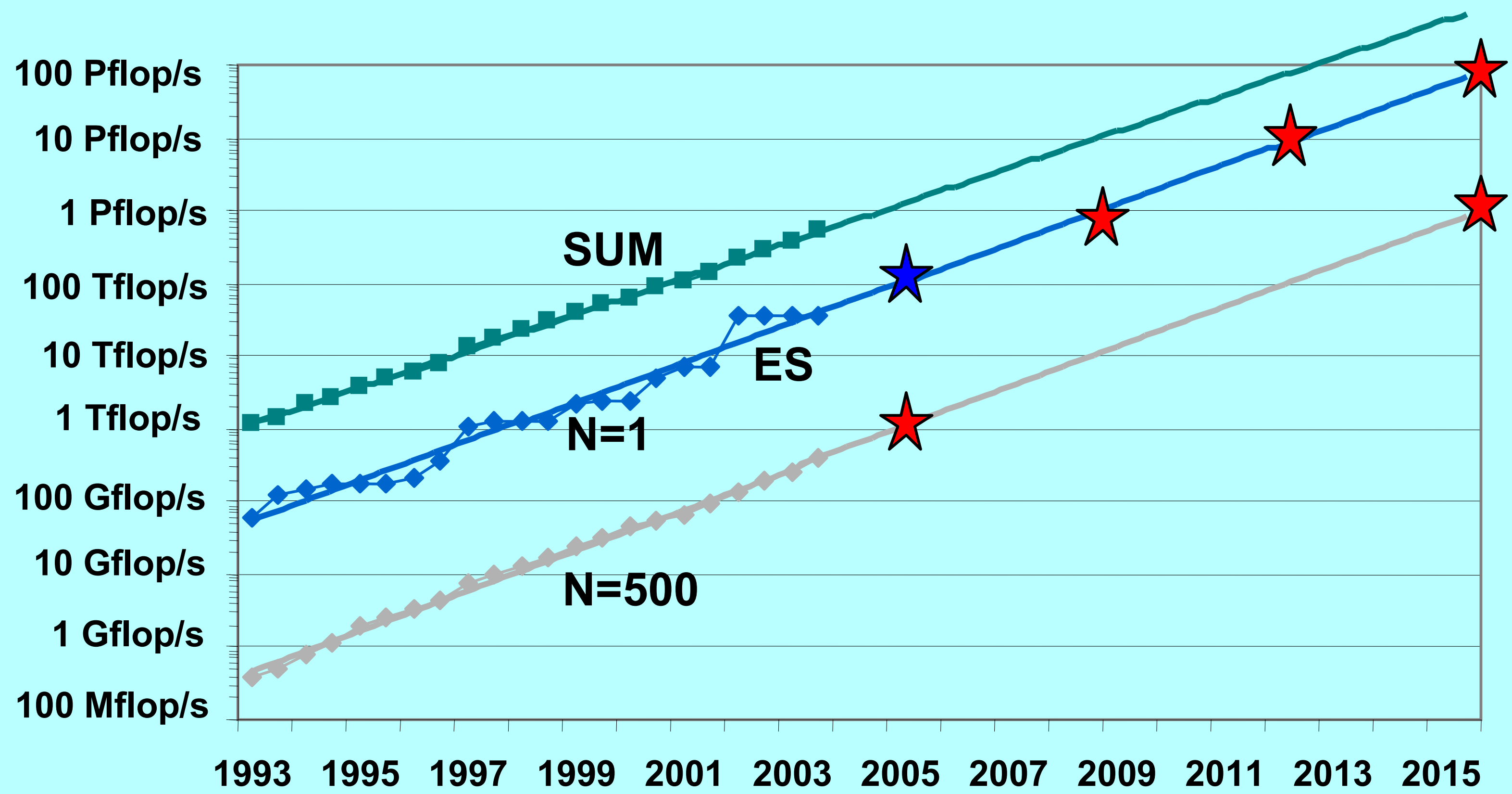
Top 10 Machines
June 2004
http://www.top500.org

Rank	Manufacturer	Computer	R_{max} [TF/s]	Installation Site	Country	Year	Area of Installation	# Proc
1	NEC	Earth-Simulator	35.86	Earth Simulator Center	Japan	2002	Research	5120
2	California Digital	Itanium 2, Quadrics	19.94	LLNL	USA	2004	Research	4096
3	HP	AlphaServer SC	13.88	LANL	USA	2002	Research	8192
4	IBM/LLNL	Blue Gene Power PC	11.68	IBM	USA	2004	Research	8192
5	Dell	PowerEdge, Myrinet	9.82	NCSA	USA	2003	Academic	2500
6	IBM	eServer Power4	8.96	ECMWF	UK	2004	Research	2112
7	Fujitsu	Fujitsu	8.73	RIKEN	Japan	2004	Research	2048
8	IBM	Blue Gene Power PC	8.67	IBM	USA	2004	Research	4096
9	HP	Itanium 2, Quadrics	8.63	PNNL	USA	2003	Research	1936
10	Dawning	Opteron, Myrinet	8.06	Shangai Supercomputing	China	2004	Research	2560

A computation that took 1 full year to complete in 1980 could be done in ~ 10 hours in 1992, in ~ 16 minutes in 1997 and in ~ 27 seconds in 2001!



Projected Performance Development



Courtesy of Erich Strohmaier

The Reality…

- Research in computational sciences is fundamentally interdisciplinary and addresses, among many others, climate and environment modeling, DNA sequencing, flows in geological structures, etc.
- In 1999, the PITAC Report recommended the creation of a national library of certified domain-specific software in order to reduce the labor required for software development, testing and evolution.
- The development of complex simulation codes on high-end computers is not a trivial task.
- Usually, a significant percentage of the efforts focus on the development of the codes and their optimization.
- There is a need for a collaboration framework for ongoing development and deployment of computational tools.

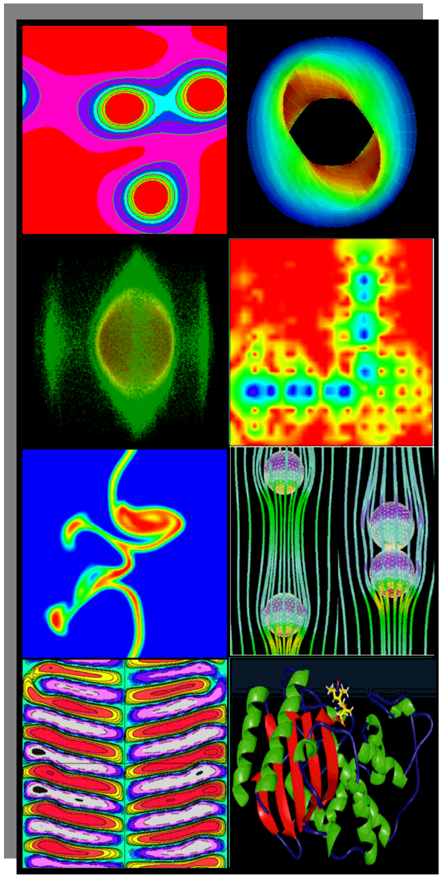
See SCaLes (Science Case for Large-scale Simulation) and HECRTF (High Computing Revitalization Task Force) reports.

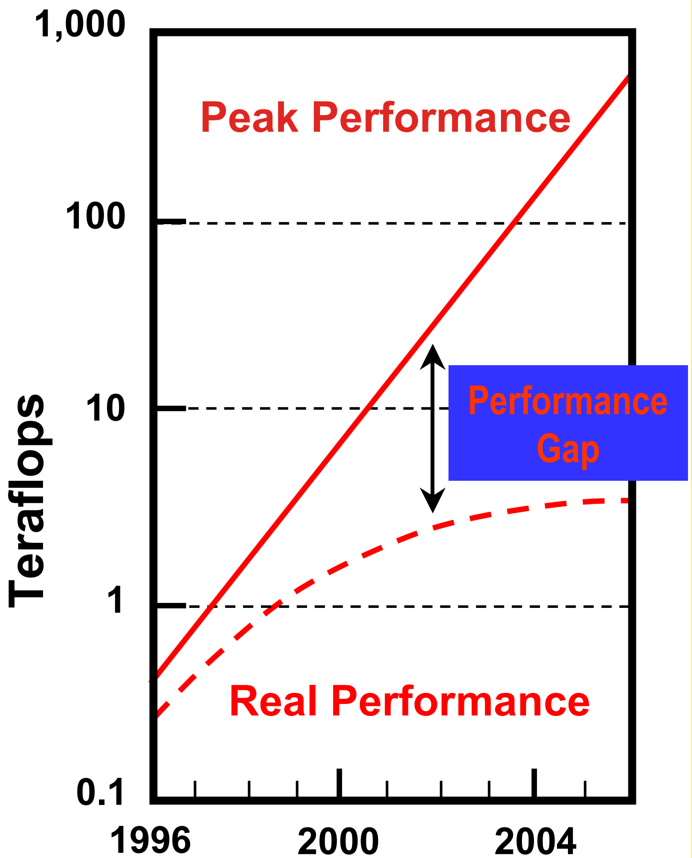
Challenges in the Development of Scientific Codes

- Productivity
  - Time to the first solution (prototype)
  - Time to solution (production)
  - Other requirements
- Complexity
  - Increasingly sophisticated models
  - Model coupling
  - Interdisciplinarity
- Performance
  - Increasingly complex algorithms
  - Increasingly complex architectures
  - Increasingly demanding applications

- Libraries written in different languages.
- Discussions about standardizing interfaces are often sidetracked into implementation issues.
- Difficulties managing multiple libraries developed by third-parties.
- Need to use more than one language in one application.
- The code is long-lived and different pieces evolve at different rates
- Swapping competing implementations of the same idea and testing without modifying the code
- Need to compose an application with some other(s) that were not originally designed to be combined

Addressing the Performance Gap through Software





Peak performance is skyrocketing

- In 1990s, peak performance increased 100x; in 2000s, it will increase 1000x

But

- Efficiency for many science applications declined from 40-50% on the vector supercomputers of 1990s to as little as 5-10% on parallel supercomputers of today

Need research on

- Mathematical methods and algorithms that achieve high performance on a single processor and scale to thousands of processors
- More efficient programming models for massively parallel supercomputers

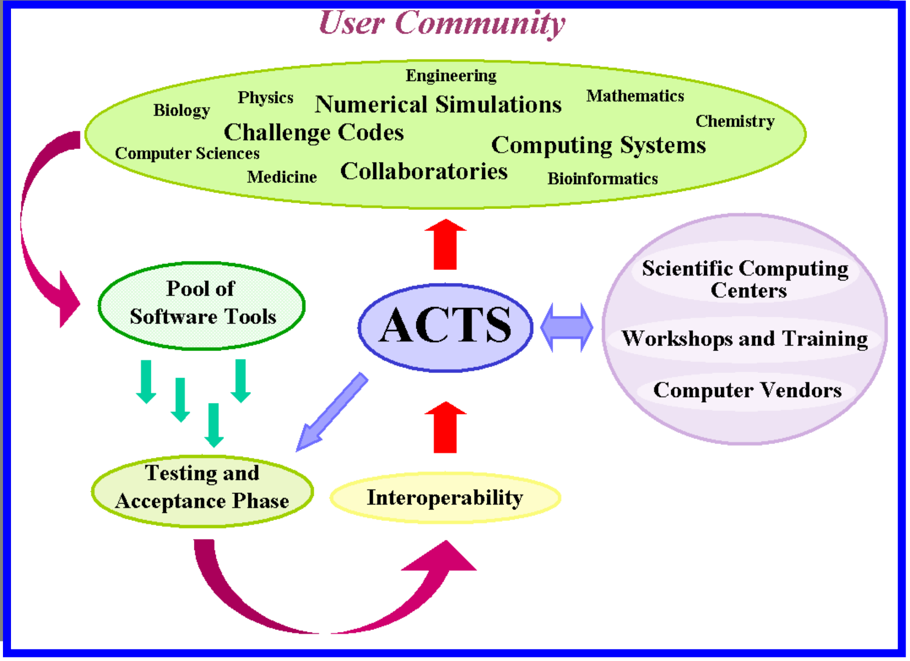
Lessons Learned

- *There is still a gap between tool developers and application developers which leads to duplication of efforts.*
- *The tools currently included in the ACTS Collection should be seen as dynamical configurable toolkits and should be grouped into toolkits upon user/application demand.*
- *Users demand long-term support of the tools.*
- *Applications and users play an important role in making the tools mature.*
- *Tools evolve or are superseded by other tools.*
- *There is a demand for tool interoperability and more uniformity in the documentation and user interfaces.*
- *There is a need for an intelligent and dynamic catalog/repository of high performance tools.*

Why is ACTS unique?

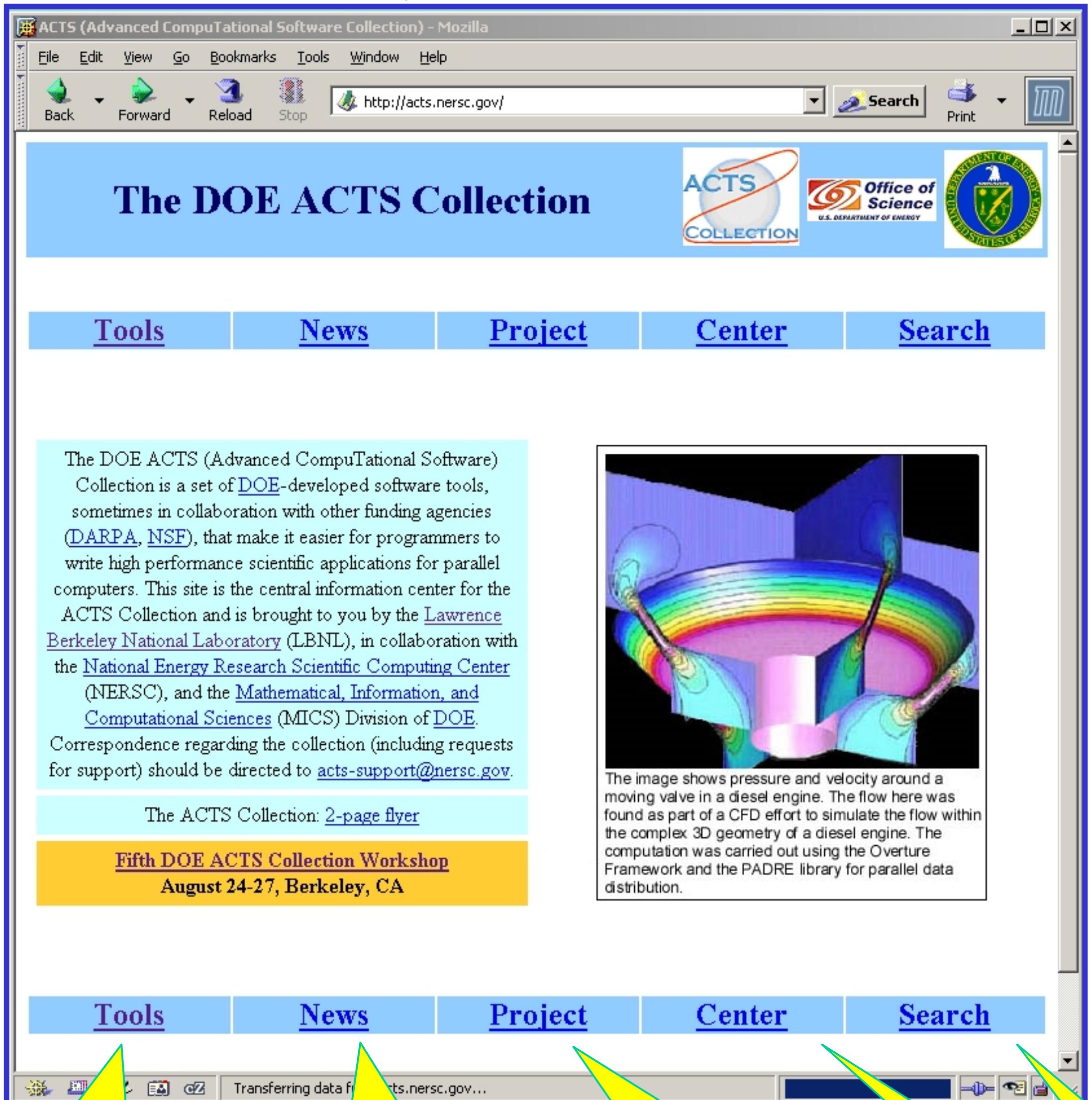
- Provides pointers and documentation about software tools.
- Accumulates the expertise and user feedback on the use of the software tools and scientific applications that used them:
  - independent software evaluations
  - participation in the developer user groups e-mail list
  - leverage between tool developers and tool users
  - presentation of a gallery of applications
  - workshops and tutorials
  - tool classification
  - support

4th ACTS Workshop, August 5-8, 2004, Berkeley, CA



http://acts.nersc.gov

See also: _http://acts.nersc.gov/documents_



Tool descriptions, installation details, examples, etc

Agenda, accomplishments, conferences, releases, etc

Goals and other relevant information

Points of contact

Search engine

- High Performance Tools
  - portable
  - library calls
  - robust algorithms
  - help code optimization
- Scientific Computing Centers (like NERSC, PSC):
  - Reduce user's code development time that sums up in more production runs and faster and effective scientific research results
  - Overall better system utilization
  - Facilitate the accumulation and distribution of high performance computing expertise
  - Provide better scientific parameters for procurement and characterization of specific user needs

Have a nice Workshop!